Amendments to the Specification:

Please replace the paragraph beginning at line 19 of page 15 with the following amended paragraph:

Computer 16, which contains the client process 11A for the backup and restore software, also has its own software modules in the preferred embodiment of the invention. The backup and restore application, has to be capable of handling either database backups of file system backups. For example, the backup and restore software has to be capable of handling backups from an Oracle or other databases (e.g. Sybase, Informix) or backups from a file system such as that which would reside on an NT server. Thus, computer 16 could potentially have software for either file systems shown at 51 or database software, such as that produced and sold by Oracle, as shown at 53. 11A shows the client process of the backup and restore software. Similar to the software residing in server process, the client process also contains the communication mechanism 58 to facilitate communication over the network 18 to the server process 11 residing on computer 12. In the preferred embodiment of the invention, the communication mechanisms 58 are TCP/IP sockets. Client process 11A, similar to server process 11, also wishes to facilitate communication to server process 11 through data storage system 14. It does this with the use of communication mechanism 60. In the preferred embodiment of the invention, communication mechanism 60 is a second set of storage sockets specially designed to facilitate communication through data storage system 14. In the preferred embodiment of the invention, the storage sockets 60 are STP or SSLsockets, as described earlier.

Please replace the paragraph beginning at line 3 of page 20 with the following amended paragraph:

It should be noted that, once the client process sends the file descriptor to the server process at step 122, the client process falls into a loop. The client process is then ready to accept another connection. Once the server process receives the file descriptor, it [[opens]] creates another socket on the same WKP. The numbers contained with the socket are simply to indicate different sockets. The other socket is ~~opened~~ created at step 94. As previously shown, the socket is followed by bind and connect commands at steps 96 and 98 respectively. The client accepts the connect at step 98 at step 124. It should be noted that all of this is done on the WKP. Once the client process has accepted the connection request at step 124, the server process sends to the client process, at step 100, the file descriptor, which is received by the client process at step 126. Step 100 allows the server to send the file descriptor to the client to let the client know it is in connection with the same server process. The second of the two connections the client process has been told by the configuration file that it going to receive is now completed. Because two connections have been made, the server process and the client process in essence have two separate channels for use in communication with each other. Typically, one of the communication channels is used for data communications, while the second is used for error communication.